**ADGORITMO DEFINICION:**

Un Algoritmo es una serie ordenada de instrucciones, pasos o procesos que llevan a la solución de un determinado problema. Los hay tan sencillos y cotidianos como seguir la receta del médico, abrir una puerta, lavarse las manos, etc; hasta los que conducen a la solución de problemas muy complejos.

Los Algoritmos permiten describir claramente una serie de instrucciones que debe realizar el computador para lograr un resultado previsible. Vale la pena recordar que un procedimiento de computador consiste de una serie de instrucciones muy precisas y escritas en un lenguaje de programación que el computador entienda, en este curso utilizaremos el entorno de desarrollo de Scratch para programar nuestros algoritmos.

En la naturaleza hay muchos procesos que puedes considerar como Algoritmos ya que tienen procedimientos y reglas. Incluso, muchas veces no somos conscientes de ellos.  
Por ejemplo, el proceso digestivo es un concepto de algoritmo con el que convivimos a diario sin que nos haga falta una definición precisa de este proceso. El hecho de que conozcamos cómo funciona el sistema digestivo, no implica que los alimentos que consumimos nos alimenten más o menos. La familiaridad de lo que sucede día a día nos impide ver muchos algoritmos que pasan a nuestro alrededor. Procesos naturales como la gestación, las estaciones, la circulación sanguínea, los ciclos planetarios, etc, son algoritmos naturales que

generalmente pasan desapercibidos.

**CARACTERISTICAS:**

* Un algoritmo **debe ser preciso** e indicar el orden de realización de cada paso.
* Un algoritmo **debe estar definido**. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
* Un algoritmo **debe ser finito**. el algoritmo se debe terminar en algún momento; o sea, debe tener un número finito de pasos
* Un algoritmo**debe ser legibles:**El texto que lo describe debe ser claro, tal que permita entenderlo y leerlo fácilmente.

**METODOS DE ALGORITMO:**

FORMAS DE REPRESENTAR UN ALGORITMO

SEUDOCÓDIGO: es una forma de diagramar o representar un algoritmo para resolver un determinado problema, independiente de cualquier lenguaje de programación en especial.  
  
DIAGRAMA DE FLUJO: es la esquematización gráfica de un algoritmo, el cual muestra gráficamente los pasos o procesos a seguir para alcanzar la solución de un problema. Es la representación de la solución utilizando figuras geométricas como circunferencias, flechas, rectángulos, rombos, etc.  
  
DIAGRAMA N - S (Nassi-Shneiderman): también conocido como diagrama de Chapin es una técnica de especificación de algoritmos que combina la descripción textual, propia del seudocódigo, con la representación gráfica del diagrama de flujo.

**DIAGRAMA DE FLUJO:**

El diagrama de flujo o diagrama de actividades es la [representación gráfica](http://es.wikipedia.org/wiki/Gr%C3%A1fica) del [algoritmo](http://es.wikipedia.org/wiki/Algoritmo) o proceso. Se utiliza en disciplinas como [programación](http://es.wikipedia.org/wiki/Programaci%C3%B3n), [economía](http://es.wikipedia.org/wiki/Econom%C3%ADa), [procesos industriales](http://es.wikipedia.org/wiki/Proceso_de_fabricaci%C3%B3n) y [psicología cognitiva](http://es.wikipedia.org/wiki/Psicolog%C3%ADa_cognitiva).

En [Lenguaje Unificado de Modelado](http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado) (UML), un diagrama de actividades representa los [flujos de trabajo](http://es.wikipedia.org/wiki/Flujo_de_trabajo) paso a paso de negocio y operacionales de los componentes en un sistema. Un diagrama de actividades muestra el flujo de control general.

En [SysML](http://es.wikipedia.org/wiki/SysML" \o "SysML) el diagrama ha sido extendido para indicar flujos entre pasos que mueven elementos físicos (p.ej., gasolina) o energía (p.ej., presión). Los cambios adicionales permiten al diagrama soportar mejor flujos de comportamiento y datos continuos.

Estos diagramas utilizan símbolos con significados definidos que representan los pasos del algoritmo, y representan el flujo de ejecución mediante flechas que conectan los puntos de inicio y de fin de proceso.

**SIMBOLOS Y NOMBRES:**

* [**Óvalo**](http://es.wikipedia.org/wiki/%C3%93valo) o [**Elipse**](http://es.wikipedia.org/wiki/Elipse): Inicio y término (Abre y cierra el diagrama).
* [**Rectángulo**](http://es.wikipedia.org/wiki/Rect%C3%A1ngulo): Actividad (Representa la ejecución de una o más actividades o procedimientos).
* [**Rombo**](http://es.wikipedia.org/wiki/Rombo): Decisión (Formula una pregunta o cuestión).
* [**Círculo**](http://es.wikipedia.org/wiki/C%C3%ADrculo): Conector (Representa el enlace de actividades con otra dentro de un procedimiento).
* [**Triángulo**](http://es.wikipedia.org/wiki/Tri%C3%A1ngulo)**boca abajo**: Archivo definitivo (Guarda un documento en forma permanente).
* **Triángulo boca arriba**: Archivo temporal (Proporciona un tiempo para el almacenamiento del documento).

**CLASES DE ALGORITMOS Y EJEMPLOS:**

En Ciencias de la computación un algoritmo es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generan duda a quien deba realizar dicha actividad. dados un estado inicial y un entrada, siguiendo los pasos sucesivos se llega a un estado final y se obtiene un solución.

Lo algoritmos son el objeto de estudio de la algoritmia los diagramas de flujo, como su nombre lo indica son gráficas que representan la dirección que sigue la información que contiene un algoritmo; los datos se encierran en diferentes figuras, estas se llaman figuras lógicas. Existen 5 figuras lógicas únicas utilizadas en el diagrama de fuljo: inicio, proceso, pregunta, ciclo y fin.

El algoritmo es de carácter general y puede aplicarse a cualquier información matemática o a cualquier problema.

La formulación del algoritmo fue uno de las mas grandes adelantos dentro de la ciencia matemática ya que debe partir de ello se pudieron resolver infinidad de problemas.

Los algoritmos para llegar a ser tales deben reunir ciertas características. Una de ellas es que los pasos que deben seguirse deben estar estrictamente descritos, cada acción debe ser precisa, debe ser general, es decir, que puede ser aplicable a todos los elementos de una misma clase.

**VARIABLES**

En [programación](http://es.wikipedia.org/wiki/Programaci%C3%B3n), una variable está formada por un espacio en el sistema de almacenaje ([memoria principal](http://es.wikipedia.org/wiki/Memoria_principal) de un [ordenador](http://es.wikipedia.org/wiki/Ordenador)) y un nombre simbólico (un [identificador](http://es.wikipedia.org/wiki/Identificador)) que está asociado a dicho espacio. Ese espacio contiene una cantidad o información conocida o desconocida, es decir un [valor](http://es.wikipedia.org/wiki/Valor_(inform%C3%A1tica)). El nombre de la variable es la forma usual de [referirse](http://es.wikipedia.org/wiki/Referencia_(inform%C3%A1tica)) al valor almacenado: esta separación entre nombre y contenido permite que el nombre sea usado independientemente de la información exacta que representa. El identificador, en el [codigo fuente](http://es.wikipedia.org/wiki/Codigo_fuente" \o "Codigo fuente) de la computadora puede estar [ligado](http://es.wikipedia.org/wiki/Binding) a un valor durante el [tiempo de ejecución](http://es.wikipedia.org/wiki/Tiempo_de_ejecuci%C3%B3n) y el valor de la variable puede por lo tanto cambiar durante el curso de la ejecución del programa. El concepto de variables en computación puede no corresponder directamente al concepto de [variables en matemática](http://es.wikipedia.org/wiki/Variable_(matem%C3%A1ticas)). El valor de una variable en computación no es necesariamente parte de una [ecuación](http://es.wikipedia.org/wiki/Ecuaci%C3%B3n) o [fórmula](http://es.wikipedia.org/wiki/F%C3%B3rmula) como en matemáticas. En computación una variable puede ser utilizada en un proceso repetitivo: puede asignársele un valor en un sitio, ser luego utilizada en otro, más adelante reasignársele un nuevo valor para más tarde utilizarla de la misma manera.Procedimientos de este tipo son conocidos con el nombre de [iteración](http://es.wikipedia.org/wiki/Iteraci%C3%B3n). En programación de computadoras, a las variables, frecuentemente se le asignan nombres largos para hacerlos relativamente descriptivas para su uso, mientras que las variables en matemáticas a menudo tienen nombres escuetos, formados por uno o dos caracteres para hacer breve en su transcripción y manipulación.

**VALOR DE LAS VARIABLES:**

Las variables pueden ser de longitud:

* Fija.: Cuando el tamaño de la misma no variará a lo largo de la ejecución del programa. Todas las variables, sean del [tipo](http://es.wikipedia.org/wiki/Tipo_de_dato) que sean tienen longitud fija, salvo algunas excepciones como las colecciones de otras variables ([arrays](http://es.wikipedia.org/wiki/Array" \o "Array)) o las [cadenas](http://es.wikipedia.org/wiki/Cadena_(inform%C3%A1tica)).
* Variable: Cuando el tamaño de la misma puede variar a lo largo de la ejecución. Típicamente colecciones de datos.

**CONSTANTES :**

En [programación](http://es.wikipedia.org/wiki/Programaci%C3%B3n), una constante es un valor que no puede ser alterado/modificado durante la ejecución de un programa, únicamente puede ser leído.

Una constante corresponde a una longitud fija de un área reservada en la memoria principal del ordenador, donde el programa almacena valores fijos.

Por ejemplo:

* El valor de pi = 3.1416

Por conveniencia, el nombre de las constantes suele escribirse en mayúsculas en la mayoría de lenguajes.

**PSEUDOCODIGO:**

En [ciencias de la computación](http://es.wikipedia.org/wiki/Ciencias_de_la_computaci%C3%B3n), y [análisis numérico](http://es.wikipedia.org/wiki/An%C3%A1lisis_num%C3%A9rico), el pseudocódigo (o falso [lenguaje](http://es.wikipedia.org/wiki/Lenguaje)) es una descripción de [alto nivel](http://es.wikipedia.org/wiki/Lenguaje_de_alto_nivel) compacta e informal[1](http://es.wikipedia.org/wiki/Pseudoc%C3%B3digo#cite_note-1) del principio operativo de un[programa informático](http://es.wikipedia.org/wiki/Programa_inform%C3%A1tico) u otro [algoritmo](http://es.wikipedia.org/wiki/Algoritmo).

Utiliza las convenciones estructurales de un [lenguaje de programación](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) real,[2](http://es.wikipedia.org/wiki/Pseudoc%C3%B3digo#cite_note-2) pero está diseñado para la lectura humana en lug

Dar de la lectura mediante máquina, y con independencia de cualquier otro lenguaje de programación. Normalmente, el pseudocódigo omite detalles que no son esenciales para la comprensión humana del algoritmo, tales como declaraciones de variables, código específico del sistema y algunas [subrutinas](http://es.wikipedia.org/wiki/Subrutina). El lenguaje de programación se complementa, donde sea conveniente, con descripciones detalladas en [lenguaje natural](http://es.wikipedia.org/wiki/Lenguaje_natural), o con notación matemática compacta. Se utiliza pseudocódigo pues este es más fácil de entender para las personas que el código del lenguaje de programación convencional, ya que es una descripción eficiente y con un entorno independiente de los principios fundamentales de un algoritmo. Se utiliza comúnmente en los libros de texto y publicaciones científicas que se documentan varios algoritmos, y también en la planificación del desarrollo de programas informáticos, para esbozar la estructura del programa antes de realizar la efectiva codificación.

No existe una [sintaxis](http://es.wikipedia.org/wiki/Sintaxis) estándar para el pseudocódigo, aunque los ocho [IDE](http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado" \o "Entorno de desarrollo integrado)'s que manejan pseudocódigo tengan su sintaxis propia. Aunque sea parecido, el pseudocódigo no debe confundirse con los programas esqueleto que incluyen código ficticio, que pueden ser [compilados](http://es.wikipedia.org/wiki/Compilador) sin errores. Los [diagramas de flujo](http://es.wikipedia.org/wiki/Diagrama_de_flujo) y [UML](http://es.wikipedia.org/wiki/UML) pueden ser considerados como una alternativa gráfica al pseudocódigo, aunque sean más amplios en papel.

**CARACTERISTICAS:**

Las principales características de este lenguaje son:

1. Se puede ejecutar en un [ordenador](http://es.wikipedia.org/wiki/Computadora) (con un [IDE](http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado) como por ejemplo SLE, LPP, PilatoX, Maruga Script, Seudocódigo o [PSeInt](http://es.wikipedia.org/wiki/PSeInt" \o "PSeInt). Otros Ides de consideración son Inter-P y Algor)
2. Es una forma de representación sencilla de utilizar y de manipular.
3. Facilita el paso del programa al lenguaje de programación.
4. Es independiente del lenguaje de programación que se vaya a utilizar.
5. Es un método que facilita la programación y solución al algoritmo del programa.

Todo documento en pseudocódigo debe permitir la descripción de:

1. Instrucciones primitivas.
2. Instrucciones de proceso....
3. Instrucciones de control.
4. Instrucciones compuestas.
5. Instrucciones de descripción.

Estructura a seguir en su realización:

1. Cabecera.
   1. Programa.
   2. Módulo.
   3. Tipos de datos.
   4. Constantes.
   5. Variables.
2. Cuerpo.
   1. Inicio.
   2. Instrucciones.
   3. Fin.